

## CLAIMS

What is claimed is:

1. A method comprising:  
caching a plurality of successive type hierarchy references corresponding to a data object within the data structure of the data object; and  
accessing the cached type hierarchy references at run time to perform type checking of the data object.
2. The method of claim 1 wherein the plurality of successive type hierarchy references are cached in a data structure of the data object.
3. The method of claim 2 wherein the data structure is a data structure of an object oriented computer language.
4. The method of claim 3 wherein the object oriented computer language is selected from the list consisting of JAVA, C++, C#, and CLI.
5. The method of claim 1 wherein the plurality of successive type hierarchy references comprises three successive type hierarchy references.
6. The method of claim 1 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references required by a specific application.

7. The method of claim 1 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.

8. The method of claim 1 further comprising:

- a) determining that the plurality of successive type hierarchy references is insufficient to check the type of the data object;
- b) obtaining a highest type hierarchy reference from the cache;
- c) accessing a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent plurality of cached successive type hierarchy references corresponding to the subsequent data object; and
- d) accessing the subsequent plurality of cached successive type hierarchy references; and
- e) repeating operations a) through d) such that type checking of the data object is accomplished.

9. A machine-readable medium provides executable instructions which, when executed by a processor, cause the processor to perform a method, the method comprising:

- caching a plurality of successive type hierarchy references corresponding to a data object within the data structure of the data object; and
- accessing the cached type hierarchy references at run time to perform type checking of the data object.

10. The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references are cached in a data structure of the data object.

11. The machine-readable medium of claim 10 wherein the data structure is a data structure of an object oriented computer language.

12. The machine-readable medium of claim 11 wherein the object oriented computer language is selected from the list consisting of JAVA, C++, C#, and CLI.

13. The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises three successive type hierarchy references.

14. The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references required by a specific application.

15. The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.

16. The machine-readable medium of claim 9 wherein the method further comprises:

- a) determining that the plurality of successive type hierarchy references is insufficient to check the type of the data object;
- b) obtaining a highest type hierarchy reference from the cache;

c) accessing a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent plurality of cached successive type hierarchy references corresponding to the subsequent data object; and

d) accessing the subsequent plurality of cached successive type hierarchy references; and

e) repeating operations a) through d) such that type checking of the data object is accomplished.

17. A machine-readable medium having stored thereon at least one data object, the at least one data object having a data structure comprising:

a type field to specify the data object type; and

an identifier that identifies successive type hierarchy references of the data object.

18. The machine-readable medium of claim 17 wherein the identifier is a sub-root log to store the plurality of successive type hierarchy references of the data object.

19. The machine-readable medium of claim 17 wherein the identifier is a pointer to a sub-root log, the sub-root log to store the plurality of successive type hierarchy references of the data object.

20. The machine-readable medium of claim 17 wherein the data objects are data objects of an object oriented computer language.

21. The machine-readable medium of claim 20 wherein the object oriented computer language is selected from the list consisting of JAVA, C++, C#, and CLI.

22. The machine-readable medium of claim 17 wherein the plurality of successive type hierarchy references comprises three successive type hierarchy references.

23. The machine -readable medium of claim 17 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references required by a specific application.

24. An apparatus comprising:  
a cache memory having stored therein a plurality of successive type hierarchy references corresponding to a data object;  
a main memory having stored therein instructions; and  
a processor to execute the instructions such that execution of the instructions causes the processor to access the cached type hierarchy references at run time to perform type checking of the data object.

25. The apparatus of claim 24 wherein the data object includes a data structure, the data structure storing the plurality of successive type hierarchy references.

26. The apparatus of claim 24 wherein the data object includes a data structure, the data structure storing a pointer to a sub-root log, the sub-root log storing the plurality of successive type hierarchy references.

27. The apparatus of claim 24 wherein the instructions include code of an object oriented computer language.

28. The apparatus of claim 24 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references required by a specific application.

29. The apparatus of claim 24 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.

30. The apparatus of claim 24 wherein the main memory stores further instructions such that execution of the further instructions causes the processor to:

- a) determine that the plurality of successive type hierarchy references is insufficient to check the type of the data object;
  - b) obtain a highest type hierarchy reference from the cache;
  - c) access a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent plurality of cached successive type hierarchy references corresponding to the subsequent data object;
  - d) access the subsequent plurality of cached successive type hierarchy references;
- and
- e) repeat operations a) through d) such that type checking of the data object is accomplished.